

Exhibit 1

Conditionally Filed Under Seal

Exhibit A

Conditionally Filed Under Seal

Exhibit B

Conditionally Filed Under Seal

Exhibit C

Conditionally Filed Under Seal

Exhibit D

Conditionally Filed Under Seal

Exhibit E

PRODUCTS

PRICING

SOLUTIONS

API & DOCS

HELP

SIGNUP

LOGIN

Quickstart

HowTo's

Helper Libraries

API docs

Security

Twilio REST Web Service Interface

The Twilio REST API allows you to query meta-data about your account, phone numbers, calls, text messages, and recordings. You can also do some fancy things like initiate outbound calls and send text messages.

Since the API is based on REST principles, it's very easy to write and test applications. You can use your browser to access URLs, and you can use pretty much any HTTP client in any programming language to interact with the API.

Base URL

All URLs referenced in the documentation have the following base:

`https://api.twilio.com/2010-04-01`

The Twilio REST API is served over HTTPS. To ensure data privacy, unencrypted HTTP is not supported.

Subresources

Twilio Accounts have the following subresources. Click on a link to read the API documentation for accessing or modifying each resource.

Calls

- [Make a Call](#)
- [Call a SIP-enabled endpoint](#)
- [Retrieve Call Logs](#)
- [Modify Live Calls](#)
- [Verify a Caller ID](#)
- [View Call Transcriptions](#)
- [View Call Recordings](#)
- [Manage Conferences](#)
- [View Conference Participants](#)

Call Queues

- [Create a Call Queue](#)
- [View Call Queues](#)

[Quickstart Tutorials](#)
[HowTo's and Examples](#)
[Helper Libraries](#)
[Twiml Reference](#)
[REST API Reference](#)
[Version 2010-04-01](#)
[Overview](#)
[Request Formats](#)
[Response Formats](#)
[Test Credentials](#)
[REST Resources](#)
[Accounts](#)
[Subaccounts](#)
[AvailablePhoneNumbers](#)
[OutgoingCallerIds](#)
[IncomingPhoneNumbers](#)
[Applications](#)
[ConnectApps](#)
[AuthorizedConnectApps](#)
[Calls](#)
[Making Calls](#)
[Modifying Live Calls](#)
[Making Calls via SIP](#)
[Call Feedback](#)
[Conferences](#)
[Participants](#)
[Queues](#)
[Members](#)
[Short Codes](#)
[Messages](#)
[Sending SMS and MMS](#)
[Media](#)
[Recordings](#)
[Transcriptions](#)
[Notifications](#)
[SIP](#)
[Domains](#)
[IpAccessControlLists](#)
[CredentialLists](#)

- View Queue Members
- Dequeue a Member

Messaging

- Send an SMS or MMS
- Send a message from a Short Code
- Retrieve Message Logs

Phone Numbers

- Search for Phone Numbers
- Purchase Phone Numbers
- Update Phone Number Properties

Usage

- View Usage Data for an Account
- Set Triggers for Usage Thresholds

Accounts

- Accounts
- Subaccounts

Applications

- Applications
- Connect Apps
- Authorized Connect Apps

SIP

- Domains
- IP Access Control Lists
- Credential Lists

Miscellaneous

- Error Notifications
- Sandbox

About REST (REpresentational State Transfer)

We designed the Twilio API in a very RESTful way, so that your consumption of it is simple and straightforward. From Wikipedia:

REST's proponents argue that the Web's scalability and growth are a direct result of a few key design principles:

- Application state and functionality are divided into resources

Usage

Records

Triggers

Deprecated Resources

Sandbox

SMS Messages

Sending SMS

Additional Information

Tips and Tricks

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability

Learning Resources

- Every resource is uniquely addressable using a universal syntax for use in hypermedia links
- All resources share a uniform interface for the transfer of state between client and resource, consisting of
 - A constrained set of well-defined operations
 - A constrained set of content types, optionally supporting code on demand
- A protocol which is:
 - Client-server
 - Stateless
 - Cacheable
 - Layered

REST's client/server separation of concerns simplifies component implementation, reduces the complexity of connector semantics, improves the effectiveness of performance tuning, and increases the scalability of pure server components. Layered system constraints allow intermediaries-proxies, gateways, and firewalls-to be introduced at various points in the communication without changing the interfaces between components, thus allowing them to assist in communication translation or improve performance via large-scale, shared caching.

REST enables intermediate processing by constraining messages to be self-descriptive: interaction is stateless between requests, standard methods and media types are used to indicate semantics and exchange information, and responses explicitly indicate cacheability.

If you're looking for more information about RESTful web services, the O'Reilly RESTful Web Services book is excellent.

COMPANY

About Us
Team
Jobs

PRODUCTS

Voice
Client
Messaging

RESOURCES

API & Docs
Security
International
Open Source

COMMUNITY

Twilio Blog
Engineering Blog
Twilio.org

SOLUTIONS

Elements
Showcase
Partners
DCers

SUPPORT

Help Center
Status
Talk to Sales

Exhibit F

PRODUCTS

PRICING

SOLUTIONS

API & DOCS

HELP

SIGNUP

LOGIN

Quickstart

HowTo's

Helper Libraries

API docs

Security

PHP

REST API: Your Request

Credentials

All requests to Twilio's REST API require you to authenticate using HTTP basic auth to convey your identity. The username is your AccountSid (a 34 character string, starting with the letters AC). The password is your AuthToken. Your AccountSid and AuthToken are on the Account Dashboard page.

Most HTTP clients (including web-browsers) present a dialog or prompt for you to provide a username and password for HTTP basic auth. Most clients will also allow you to provide credentials in the URL itself. For example:

```
https://{AccountSid}:{AuthToken}@api.twilio.com/2010-04-01/Accounts
```

Retrieving Resources with the HTTP GET Method

You can retrieve a representation of a resource by GETting its url. The easiest way to do this is to copy and paste a URL into your web browser's address bar.

Possible GET Response Status Codes

- **200 OK:** The request was successful and the response body contains the representation requested.
- **302 FOUND:** A common redirect response; you can GET the representation at the URI in the Location response header.
- **304 NOT MODIFIED:** Your client's cached version of the representation is still up to date.
- **401 UNAUTHORIZED:** The supplied credentials, if any, are not sufficient to access the resource.
- **404 NOT FOUND:** You know this one.
- **429 TOO MANY REQUESTS:** Your application is sending too many simultaneous requests.
- **500 SERVER ERROR:** We couldn't return the representation due to an internal server error.
- **503 SERVICE UNAVAILABLE:** We are temporarily unable to return the representation. Please wait for a bit and try again.

Quickstart Tutorials

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

Version 2010-04-01

Overview

Request Formats

Response Formats

Test Credentials

REST Resources

Accounts

Subaccounts

AvailablePhoneNumbers

OutgoingCallerIds

IncomingPhoneNumbers

Applications

ConnectApps

AuthorizedConnectApps

Calls

Making Calls

Modifying Live Calls

Making Calls via SIP

Call Feedback

Conferences

Participants

Queues

Members

Short Codes

Messages

Sending SMS and MMS

Media

Recordings

Transcriptions

Notifications

SIP

Domains

IpAccessControlLists

CredentialLists

Creating or Updating Resources with the HTTP POST and PUT Methods

Creating or updating a resource involves performing an HTTP PUT or HTTP POST to a resource URL. In the PUT or POST, you represent the properties of the object you wish to update as form urlencoded key/value pairs. Don't worry, this is already the way browsers encode POSTs by default. But be sure to set the HTTP Content-Type header to "application/x-www-form-urlencoded" for your requests if you are writing your own client.

Possible POST or PUT Response Status Codes

- 200 OK: The request was successful, we updated the resource and the response body contains the representation.
- 201 CREATED: The request was successful, we created a new resource and the response body contains the representation.
- 400 BAD REQUEST: The data given in the POST or PUT failed validation. Inspect the response body for details.
- 401 UNAUTHORIZED: The supplied credentials, if any, are not sufficient to create or update the resource.
- 404 NOT FOUND: You know this one.
- 405 METHOD NOT ALLOWED: You can't POST or PUT to the resource.
- 429 TOO MANY REQUESTS: Your application is sending too many simultaneous requests.
- 500 SERVER ERROR: We couldn't create or update the resource. Please try again.

Deleting Resources with the HTTP DELETE Method

To delete a resource make an HTTP DELETE request to the resource's URL. Not all Twilio REST API resources support DELETE.

Possible DELETE Response Status Codes

- 204 OK: The request was successful; the resource was deleted.
- 401 UNAUTHORIZED: The supplied credentials, if any, are not sufficient to delete the resource.
- 404 NOT FOUND: You know this one.
- 405 METHOD NOT ALLOWED: You can't DELETE the resource.
- 429 TOO MANY REQUESTS: Your application is sending too many simultaneous requests.
- 500 SERVER ERROR: We couldn't delete the resource. Please try again.

HTTP Method Overloading

Twilio's REST API uses HTTP GET, POST, PUT and DELETE methods. Since some HTTP clients do not support methods PUT and DELETE, you can simulate them via POST by appending the query string parameter `_method` (yes, underscore method) to a resource URL. Valid values are PUT and DELETE.

Usage
Records
Triggers
Deprecated Resources
Sandbox
SMS Messages
Sending SMS
Additional Information
Tips and Tricks

SIP Reference
Client Reference
Connect Reference
Debugging Errors
Security
Availability & Reliability
Learning Resources

For example, if you want to perform a DELETE request on a particular phone number resource you could:

```
DELETE /2010-04-01/Accounts/AC30947.../IncomingPhoneNumbers/PN12345567
```

But if your client is only capable of GET and POST, then you could perform a POST with a `_method` query string variable to achieve the same result:

```
POST /2010-04-01/Accounts/AC30947.../IncomingPhoneNumbers/PN12345567?_method=DELETE
```

COMPANY

About Us
Team
Jobs

PRODUCTS

Voice
Client
Messaging

RESOURCES

API & Docs
Security
International
Open Source

COMMUNITY

Twilio Blog
Engineering Blog
Twilio.org

SOLUTIONS

Elements
Showcase
Partners
DOers

SUPPORT

Help Center
Status
Talk to Sales

© 2009 - 2014 Twilio, Inc. All rights reserved. Patents Pending.

[Terms of Service](#) | [Privacy Policy](#)

Exhibit G

PRODUCTS

PRICING

SOLUTIONS

API & DOCS

HELP

SIGNUP

LOGIN

Quickstart

HowTo's

Helper Libraries

API docs

Security

PHP

REST API: Accounts

When you first sign up with Twilio, you have just one account, your Master account. But you can also create more accounts... subaccounts are useful for things like segmenting phone numbers and usage data for your customers and controlling access to data. For more information on subaccounts see Using Subaccounts.

Account Instance Resource

An Account instance resource represents a single Twilio account.

Resource URI

`/2010-04-01/Accounts/{AccountSid}`

Resource Properties

An Account resource is represented by the following properties:

Property	Description
Sid	A 34 character string that uniquely identifies this account.
DateCreated	The date that this account was created, in GMT in RFC 2822 format
DateUpdated	The date that this account was last updated, in GMT in RFC 2822 format.
FriendlyName	A human readable description of this account, up to 64 characters long. By default the FriendlyName is your email address.
Type	The type of this account. Either <code>trial</code> or <code>full</code> if you've upgraded.
Status	The status of this account. Usually <code>active</code> , but can be <code>suspended</code> or <code>closed</code> .
AuthToken	The authorization token for this account. This token should be kept a secret, so no sharing.
Uri	The URI for this resource, relative to <code>https://api.twilio.com</code> .
SubresourceUris	The list of subresources under this account.
OwnerAccountSid	The Sid of the parent account for this account. The <code>OwnerAccountSid</code> of a parent account is its own sid.

Quickstart Tutorials

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

Version 2010-04-01

Overview

Request Formats

Response Formats

Test Credentials

REST Resources

Accounts

Subaccounts

AvailablePhoneNumbers

OutgoingCallerIds

IncomingPhoneNumbers

Applications

ConnectApps

AuthorizedConnectApps

Calls

Making Calls

Modifying Live Calls

Making Calls via SIP

Call Feedback

Conferences

Participants

Queues

Members

Short Codes

Messages

Sending SMS and MMS

Media

Recordings

Transcriptions

Notifications

SIP

Domains

IpAccessControlLists

CredentialLists

HTTP GET

Returns a representation of an account, including the properties above.

Example

JSON XML PHP Python C# Java Ruby Node.js

```

1 <?php
2 // Get the PHP helper Library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the L
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "ACba8bc05eacf94afdae398e642c9cc32d";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Get an object from its sid. If you do not have a sid,
11 // check out the list resource examples on this page
12 $account = $client->accounts->get("ACba8bc05eacf94afdae398e642c9cc32d");
13 echo $account->date_created;

```

HTTP POST and PUT

Allows you to modify the properties of an account.

See the Subaccounts reference for more information on suspending, unsuspending or closing subaccounts using the 'Status' parameter.

Optional Parameters

You may POST the following parameters:

Property	Description
FriendlyName	Update the human-readable description of this account.
Status	Alter the status of this account: use <code>closed</code> to irreversibly close this account, <code>suspended</code> to temporarily suspend it, or <code>active</code> to reactivate it.

Example 1

Suspend a subaccount by POSTing 'Status' = 'suspended':

JSON XML PHP Python C# Java Ruby Node.js

```

1 <?php
2 // Get the PHP helper Library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the L
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "AC1365ff479ef6502d85c27be6467a310c";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Get an object from its sid. If you do not have a sid,
11 // check out the list resource examples on this page
12 $account = $client->accounts->get("AC1365ff479ef6502d85c27be6467a310c");
13 $account->update(array(
14     "Status" => "suspended"
15 ));
16 echo $account->date_created;

```

Usage

Records

Triggers

Deprecated Resources

Sandbox

SMS Messages

Sending SMS

Additional Information

Tips and Tricks

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability

Learning Resources

Example 2

Re-activate a subaccount that was previously suspended by POSTing 'Status' = 'active':

JSON XML PHP Python C# Java Ruby Node.js

```

1 <?php
2 // Get the PHP helper Library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the L
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "AC1365ff479ef6502d85c27be6467a310c";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Get an object from its sid. If you do not have a sid,
11 // check out the list resource examples on this page
12 $account = $client->accounts->get("AC1365ff479ef6502d85c27be6467a310c");
13 $account->update(array(
14     "Status" => "active"
15 ));
16 echo $account->status;
```

Example 3

Permanently close a subaccount by POSTing 'Status' = 'closed':

JSON XML PHP Python C# Java Ruby Node.js

```

1 <?php
2 // Get the PHP helper Library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the L
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "AC12345678123456781234567812345678";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Get an object from its sid. If you do not have a sid,
11 // check out the list resource examples on this page
12 $account = $client->accounts->get("AC12345678123456781234567812345678");
13 $account->update(array(
14     "Status" => "closed"
15 ));
16 echo $account->date_created;
```

HTTP DELETE

Not supported. You can't delete an account using the REST API.

Accounts List Resource

The Accounts list resource represents the set of Accounts belonging to the Account used to make the API request. This list includes that account, along with any subaccounts belonging to it.

You can use the Accounts list resource to create subaccounts and retrieve the subaccounts that exist under your main account. See Subaccounts for more information.

Resource URI

/2010-04-01/Accounts

HTTP GET

Retrieve a list of the Account resources belonging to the account used to make the API request. This list will include that Account as well.

List Filters

The following query string parameters allow you to limit the list returned. Note, parameters are case-sensitive:

Parameter	Description
FriendlyName	Only return the Account resources with friendly names that exactly match this name.
Status	Only return Account resources with the given status. Can be closed, suspended OR active.

Example 1

List all accounts:

[JSON](#)
[XML](#)
[PHP](#)
[Python](#)
[C#](#)
[Java](#)
[Ruby](#)
[Node.js](#)

```

1 <?php
2 // Get the PHP helper library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the l
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "AC32a3c49700934481add5ce1659f04d2";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Loop over the list of accounts and echo a property for each one
11 foreach ($client->accounts as $account) {
12     echo $account->date_created;
13 }
```

Example 2

List all accounts that are currently 'active':

[JSON](#)
[XML](#)
[PHP](#)
[Python](#)
[C#](#)
[Java](#)
[Ruby](#)
[Node.js](#)

```

1 <?php
2 // Get the PHP helper Library from twilio.com/docs/php/install
3 require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the l
4
5 // Your Account Sid and Auth Token from twilio.com/user/account
6 $sid = "AC32a3c49700934481add5ce1659f04d2";
7 $token = "{{ auth_token }}";
8 $client = new Services_Twilio($sid, $token);
9
10 // Loop over the list of accounts and echo a property for each one
11 foreach ($client->accounts->getIterator(0, 50; array(
12     "Status" => "active"
13 )) as $account
14 ) {
15     echo $account->friendly_name;
16 }
```

HTTP POST

Create a new Account instance resource as a subaccount of the one used to make the request. See [Creating Subaccounts](#) for more information.

HTTP PUT

Not supported

HTTP DELETE

Not Supported

COMPANY

[About Us](#)
[Team](#)
[Jobs](#)

PRODUCTS

[Voice](#)
[Client](#)
[Messaging](#)

RESOURCES

[API & Docs](#)
[Security](#)
[International](#)
[Open Source](#)

COMMUNITY

[Twilio Blog](#)
[Engineering Blog](#)
[Twilio.org](#)

SOLUTIONS

[Elements](#)
[Showcase](#)
[Partners](#)
[DOers](#)

SUPPORT

[Help Center](#)
[Status](#)
[Talk to Sales](#)

Exhibit H

PRODUCTS

PRICING

SOLUTIONS

API & DOCS

HELP

SIGNUP

LOGIN

Quickstart

HowTo's

Helper Libraries

API docs

Security

PHP

REST API: Sending SMS or MMS

Sending an SMS or MMS is one of the most common tasks performed on the Twilio Platform. Sending a message is as simple as POST-ing to the Messages resource but since it's a common action it's worth walking through in detail below.

HTTP POST to Messages

To send a new outgoing message, make an HTTP POST to your Messages list resource URI:

```
/2010-04-01/Accounts/{AccountSid}/Messages
```

POST Parameters

Required Parameters

The following parameters are required in your POST to send the message:

Parameter	Description
From	A Twilio phone number enabled for the type of message you wish to send. Only phone numbers or short codes purchased from Twilio work here; you cannot (for example) spoof messages from your own cell phone number.
To	The destination phone number. Format with a '+' and country code e.g., +16175551212 (E.164 format). For 'To' numbers without a '+', Twilio will use the same country code as the 'From' number. Twilio will also attempt to handle locally formatted numbers for that country code (e.g. (415) 555-1212 for US, 07400123456 for GB). If you are sending to a different country than the 'From' number, you must include a '+' and the country code to ensure proper delivery.

You must also POST at least one of the following parameters:

Parameter	Description
Body	The text of the message you want to send, limited to 1600 characters.
MediaUrl	The URL of the media you wish to send out with the message. gif, png, and jpeg content is currently supported and will be formatted correctly on the recipient's device. Other types are also accepted by the API. If you wish to send more than one image in the message body, please provide multiple MediaUrl values in the POST request.

If you are sending non-BMP characters in the message Body the number of characters could be smaller than 1600. Almost all global languages are supported without the use of the non-BMP character plane.

Quickstart Tutorials

HowTo's and Examples

Helper Libraries

Twiml Reference

REST API Reference

Version 2010-04-01

Overview

Request Formats

Response Formats

Test Credentials

REST Resources

Accounts

Subaccounts

AvailablePhoneNumbers

OutgoingCallerIds

IncomingPhoneNumbers

Applications

ConnectApps

AuthorizedConnectApps

Calls

Making Calls

Modifying Live Calls

Making Calls via SIP

Call Feedback

Conferences

Participants

Queues

Members

Short Codes

Messages

Sending SMS and MMS

Media

Recordings

Transcriptions

Notifications

SIP

Domains

IpAccessControlLists

CredentialLists

Note that if you do not specify a `MediaUrl` and the body is greater than 160 characters, the message will be sent as SMS, segmented and charged accordingly.

Content types for `MediaUrl` validation are fetched via the `content-type` header at the provided URLs. If the `content-type` header does not match the media, Twilio will reject the request. Twilio supports `image/gif`, `image/png`, and `image/jpeg` mime-types and accepts many others.

If you are sending messages while in trial mode, the 'To' phone number must be verified with Twilio.

Optional Parameters

You may include the following parameters:

Parameter	Description
StatusCallback	A URL that Twilio will POST to each time your message status changes to one of the following: <code>failed</code> , <code>sent</code> , <code>delivered</code> , or <code>undelivered</code> . Twilio will POST the <code>MessageSid</code> along with the other standard request parameters as well as <code>MessageStatus</code> and <code>ErrorCode</code> .
ApplicationSid	Twilio will POST <code>MessageSid</code> as well as <code>MessageStatus=sent</code> or <code>MessageStatus=failed</code> to the URL in the <code>MessageStatusCallback</code> property of this Application. If the <code>statusCallback</code> parameter above is also passed, the Application's <code>MessageStatusCallback</code> parameter will take precedence.

StatusCallback Parameter

Each time a message status changes, Twilio will make an asynchronous HTTP request to the `StatusCallback` URL, if you provided one.

Request Parameters

The parameters Twilio passes to your application in its request to the `StatusCallback` URL include all the standard request parameters and these additional parameters:

Parameter	Description
MessageStatus	The status of the message. Message delivery information is reflected in message status. The possible values are described here.
ErrorCode	The error code (if any) associated with your message. If your message status is <code>failed</code> or <code>undelivered</code> , the <code>ErrorCode</code> can give you more information about the failure. If the message was delivered successfully, no <code>ErrorCode</code> will be present. The possible values are described here.

Example

Send an Message from 415-814-1829 to 555-867-5309 begging Jenny for a second chance including a heart image:

JSON XML PHP Python C# Java Ruby Node.js

```

1 |
2 | <?php
3 | // Get the PHP helper library from twilio.com/docs/php/install
4 | require_once('/path/to/twilio-php/Services/Twilio.php'); // Loads the L

```

Usage

Records

Triggers

Deprecated Resources

Sandbox

SMS Messages

Sending SMS

Additional Information

Tips and Tricks

SIP Reference

Client Reference

Connect Reference

Debugging Errors

Security

Availability & Reliability

Learning Resources

```

6 // Your Account Sid and Auth Token from twilio.com/user/account
7 $sid = "AC5ef8732a3c49700934481add5ce1659";
8 $token = "{{ auth_token }}";
9 $client = new Services_Twilio($sid, $token);
10 $client->account->messages->sendMessage("+14158141829", "+15558675309",

```

Rate limiting

You can queue as many messages as you like, however Twilio will only send out messages at a rate of one message per phone number per second. It is not possible to adjust this rate, and it does not vary based on the country in which your number is located.

If you anticipate the need to send out a large number of messages quickly (a time-limited promotion, for example) or at a rate greater than one message per second, you can purchase additional numbers, increasing your outbound capacity.

Short codes are not subject to the same rate limits as long-code numbers and may be a better option for you. Check out our short code FAQ to determine what is best for you.

Handling Message Replies

By specifying an Message URL for your messaging enabled Twilio phone number, Twilio will make a request to your application to notify you when someone replies to a message you send. Twilio's request and your corresponding response are covered in the Message portion of the TwiML documentation.

COMPANY

About Us
Team
Jobs

PRODUCTS

Voice
Client
Messaging

RESOURCES

API & Docs
Security
International
Open Source

COMMUNITY

Twilio Blog
Engineering Blog
Twilio.org

SOLUTIONS

Elements
Showcase
Partners
DOers

SUPPORT

Help Center
Status
Talk to Sales